

01	A	B	C	D	E
02	A	B	C	D	E
03	A	B	C	D	E
04	A	B	C	D	E
05	A	B	C	D	E
06	A	B	C	D	E
07	A	B	C	D	E
08	A	B	C	D	E
09	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E

14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E
17	A	B	C	D	E
18	A	B	C	D	E
19	A	B	C	D	E
20	A	B	C	D	E
21	A	B	C	D	E
22	A	B	C	D	E
23	A	B	C	D	E
24	A	B	C	D	E
25	A	B	C	D	E
26	A	B	C	D	E

27	A	B	C	D	E
28	A	B	C	D	E
29	A	B	C	D	E
30	A	B	C	D	E
31	A	B	C	D	E
32	A	B	C	D	E
33	A	B	C	D	E
34	A	B	C	D	E
35	A	B	C	D	E
36	A	B	C	D	E
37	A	B	C	D	E
38	A	B	C	D	E
39	A	B	C	D	E

PRACTICE QUESTIONS

College Board AP - United Kingdom

APCSA

AP Computer Science A

EDITION

Demo

QUESTIONS

9

FORMAT

Limited Edition PDF

Before you begin

What this is

9 practice questions for College Board AP - United Kingdom **APCSA**, each with its answer key and an explanation. Answer a question before you read its key — the explanation is worth more once you have committed.

If something looks wrong

Send us three things and we will check it:

- the exam code, **APCSA**
- the question number
- the email address on your order

Write to **support@mometrix.net**. We reply within one business day.

QUESTION 1

SINGLE CHOICE

Consider the following method: `public static int mystery(int[] arr, int target) { int count = 0; for (int i = 0; i < arr.length; i++) { if (arr[i] == target) { count++; } } return count; }` If the array {2, 4, 6, 4, 8, 4} is passed as arr and 4 is passed as target, what value is returned?

- ☐ A 1
- ☐ B 2
- ☒ C 3
- ☐ D 4

ANSWER C

WHY The method counts how many times target appears in arr. The value 4 appears three times (at indices 1, 3, and 5), so mystery returns 3.

QUESTION 2

SINGLE CHOICE

A class Student has a private String field name and a constructor `public Student(String n) { name = n; }`. Which of the following statements about this constructor is true?

- ☒ A It has no return type and does not return any value.
- ☐ B It must be declared with the keyword void before its name.
- ☐ C It returns the value stored in name.
- ☐ D It can only be called once per object.

ANSWER A

WHY Constructors in Java are not declared with a return type (not even void) and they do not return a value; their role is to initialize a newly created object. A constructor may be called multiple times via the new operator to create multiple Student objects.

QUESTION 3

SINGLE CHOICE

What is printed by the following code segment? `int x = 7; if (x % 2 == 0) { System.out.print("A"); } else if (x > 5) { System.out.print("B"); } else { System.out.print("C"); }`

- ☐ A
- ☒ B
- ☐ C
- ☐ D AB

ANSWER B

WHY `x % 2 == 0` is false (7 is odd), so the first branch is skipped. `x > 5` is true (`7 > 5`), so the else-if branch executes and prints B. Only one branch of an if/else-if chain executes.

QUESTION 4

SINGLE CHOICE

Consider the following code segment: `int[] nums = {3, 1, 4, 1, 5}; int temp = nums[1]; nums[1] = nums[2]; nums[2] = temp;` What are the contents of `nums` after the code executes?

- ☐ A {3, 1, 4, 1, 5}
- ☒ B {3, 4, 1, 1, 5}
- ☐ C {1, 3, 4, 1, 5}
- ☐ D {3, 1, 1, 4, 5}

ANSWER B

WHY `temp` stores `nums[1]` which is 1. Then `nums[1]` is set to `nums[2]` (4), and `nums[2]` is set to `temp` (1). The values at indices 1 and 2 are swapped, leaving {3, 4, 1, 1, 5}.

QUESTION 5

SINGLE CHOICE

What is printed by the following code segment? `ArrayList<String> words = new ArrayList<String>(); words.add("cat"); words.add("dog"); words.add(1, "bird"); words.remove(0); System.out.println(words);`

- ☒ **A** [bird, dog]
- ☐ **B** [dog, bird]
- ☐ **C** [cat, bird, dog]
- ☐ **D** [bird, dog, cat]

ANSWER A

WHY After `add("cat")` the list is [cat]. After `add("dog")` it is [cat, dog]. After `add(1, "bird")` it is [cat, bird, dog]. After `remove(0)` the element at index 0 (cat) is removed, leaving [bird, dog].

QUESTION 6

SINGLE CHOICE

A class `Book` has a private double field `price`. Which of the following is the most appropriate accessor (getter) method for this field?

- ☒ **A** `public double getPrice() { return price; }`
- ☐ **B** `public void getPrice() { System.out.println(price); }`
- ☐ **C** `public double setPrice(double p) { price = p; return price; }`
- ☐ **D** `private double getPrice() { return price; }`

ANSWER A

WHY An accessor returns the value of a private field using the field's type as the return type, takes no parameters, and is typically declared public so other classes can read the value. Option B returns void and prints, option C is a mutator pattern, and option D is not public.

QUESTION 7

SINGLE CHOICE

Given `int[][] grid = { {1, 2, 3}, {4, 5, 6} }`; what is the value of `grid[1][2]`?

- ☐ A 2
- ☐ B 3
- ☐ C 5
- ☒ D 6

ANSWER D

WHY `grid[1]` refers to the second row {4, 5, 6}. `grid[1][2]` refers to the element at index 2 of that row, which is 6.

QUESTION 8

SINGLE CHOICE

A class `Dog` extends `Animal`. Both classes have a `speak()` method (`Animal`'s returns `"..."` and `Dog`'s overrides it to return `"woof"`). Given `Animal a = new Dog();` what is printed by `System.out.println(a.speak());`?

- ☐ A ...
- ☒ B woof
- ☐ C A compile-time error occurs because `a` is declared as `Animal`.
- ☐ D A runtime exception occurs.

ANSWER B

WHY When a subclass object is referenced by a superclass variable, Java uses dynamic method dispatch: the overridden method in the actual object's class (`Dog`) is invoked at runtime, so `"woof"` is printed.

QUESTION 9

SINGLE CHOICE

A class Counter has a private int field count initialized to 0 and a public void increment() method that adds 1 to count. Which of the following is the most appropriate getter method for this class?

- ☐ A public int Counter() { return count; }
- ☒ B public int getCount() { return count; }
- ☐ C public void getCount() { return count; }
- ☐ D public static int getCount() { return count; }

ANSWER B

WHY A getter (accessor) should be public, return the type of the field (int), have the same access level as the field's visibility from outside, and be named get followed by the field name. Option A is a constructor (no return type, capitalized name). Option C returns a value from a void method, which is a compile error. Option D is static, which would not be an instance accessor for an object's field.